우선순위 지향 검색 및 적응형 버퍼관리에 의한 비디오 저장시스템

김근형[†] · 박 석[††]

요 약

비디오서버는 비디오스트림들을 저장하고 있다가 시간제약사항을 만족시키면서 보다 많은 사용자들에게 보다 신속하게 전송, 서비스할 수 있어야 한다. 본 논문에서는 비디오서버에서 빠른 응답을 유지할 뿐만 아니라 비디오스트림의 시간제약사항을 만족시키면서 지원할 수 있는 사용자들의 수를 증가시킬 수 있는 저장시스템을 제안한다. 다중 디스크환경하의 저장시스템에서 다중 비디오스트림들을 검색할 때 각 스트림들의 세그먼트들을 우선순위에 기반하여 읽음으로써 빠른 응답을 유지할 수 있을 뿐만 아니라 디스크 대역폭을 절약할 수 있다. 또한, 주어진 버퍼용량과 디스크 대역폭을 고려하면서 스트림의 세그먼트 크기를 휴리스틱하게 결정하여 버퍼를 효율적으로 활용함으로써 지원할 수 있는 사용자들의 수를 증가시킬 수 있다. 새로운 저장시스템에 대한 타당성은 이론적으로 분석되고 시뮬레이션을 통하여 평가된다.

Video Storage System based on Priority-Oriented Retrieval and Adapted Buffer Management

Keunhyung Kim[†] and Seog Park[††]

ABSTRACT

Video server which stores video streams should serve more users with the requested streams rapidly, satisfying their time constraints. In this paper, we propose the storage system that does not only maintain fast responses but also increase the number of users supported, with satisfying the time constraints characteristics of the video streams. When the video streams are retrieved in storage system under multiple disks environments, it can not only maintain the rapid responses but also can save the disk bandwidth owing to the retrieval based on priority. Also, it utilizes the given buffer capacity efficiently owing to determining the stream segment size heuristically, which is adapted to the buffer capacity. Consequentially, these policies can increase the number of users supported. The correctness of the proposed policies is not only analyzed theoretically but also evaluated through simulations.

Key words: 비디오서버, 우선순위, 비동기화, 시작지연시간, 더블버퍼링

1. 서 론

컴퓨터와 통신 기술의 발전으로 대용량의 멀티미디어 데이터를 저장, 접근, 처리하는 것이 가능하게 되었고 현실 세계에서 이를 응용하기 위한 활동들이 활발히 진행되고 있다. 특히, 비디오 데이터는 교육, 오락등과 같은 분야에서 인터넷 기술등과 연동되었을 때 상업적 잠재력이 크므로 비디오 데이터 관련연구들이 많이 이루어지고 있다.

비디오스트림의 특징은 우선 대용량이고 실시간 제약사항이 만족되어야 데이터의 일관성이 유지된다. 즉, 비디오서버의 저장시스템은 대용량의 많은 비디오스트림들을 실시간안에 검색할 수 있어야 하므로 높은 디스크 대역폭을 요구하게 되고 이를 충족시키기 위하여 일반적으로 여러 개의 디스크들로 이

† 제주대학교 경상대학 경영정보학과
†† 정회원, 서강대학교 공과대학 컴퓨터학과

루어지는 다중디스크 환경이 바람직하다. 다중디스크 환경이라 할지라도 제공되는 디스크 대역폭은 제한되어 있으므로 이를 효율적으로 활용할 필요가 있다. 디스크 상의 블록 데이터(블록 크기 << 100KB인 경우)를 읽을 때 디스크헤드가 순차적으로(sequentially) 이동하면서 읽는 경우가 임의적으로(randomly) 이동하면서 읽는 경우보다 2배 이상의 처리율 향상을 가져온다[1]. 즉, 디스크헤드의 임의접근은 탐색지연시간 등으로 디스크 대역폭을 낭비하게 되어 순차접근에 비하여 보다 낮은 디스크 대역폭을 제공한다. 따라서, 디스크헤드의 순차적 이동에 의해서 비디오스트림을 읽을 수 있게 하여 서비스 가능한 스트림 수를 많게 하는 저장시스템이 보다 효율적이라 할 수 있다. 특히, 비디오스트림은 시간제약 특성을 갖고 있기 때문에 순차접근을 하면서도 시간제약 조건을 만족시킬 수 있는 비디오스트림 스케쥴링 정책이 필요하다. 비디오서버의 또 다른 성능향상 방안으로 시스템이 현실적으로 제공하는 버퍼용량을 휴리스틱하게 활용함으로써 제한된 버퍼용량으로 보다 많은 스트림들을 서비스할 수 있는 저장시스템이 보다 효율적이라 할 수 있다. 일반적으로 VOD(Video-On-Demand)응용을 위한 비디오서버라면 비디오스트림 요구에 대하여 몇 분 정도의 시작지연(startup latency)은 허용될 수 있지만 비디오 데이터베이스와 같은 빠른 검색 결과를 요구하는 경우를 위한 비디오서버는 최대한 빠른 시작지연을 제공해야 한다[2]. 따라서, 비디오서버를 위한 저장시스템은 스트림 요구에 대한 시작지연을 가능한 한 단축시킬 수 있어야 한다.

본 논문에서는 비디오서버의 성능을 향상시키기 위하여 서비스요구의 시작지연시간을 단축시킬 뿐만 아니라 동시에 서비스 가능한 스트림 수를 많게 하기 위한 정책들을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 비디오스트림 배치 및 검색정책과 관련하여 지금까지 연구된 주요 정책들을 고찰하고 비교 분석하여 문제점을 제기한다. 3장에서 이러한 문제점들을 해결하기 위한 새로운 정책들을 제안한다. 4장에서는 모의 실험을 통하여 제안된 정책을 평가하고 분석하며, 마지막으로 5 장에서 결론을 맺는다.

2. 기존 관련연구의 고찰

저장 시스템이 비디오스트림에 물리적으로 접근하는 단위를 세그먼트라고 부른다. 즉, 스트림은 세그먼트들로 분할되어 다중 디스크들상에 분산 배치되고 스트림 검색시 세그먼트 단위로 읽어들인다[3]. 비디오스트림을 검색하여 재생할 때는(서비스할 때) 그 스트림의 연속적인 세그먼트들을 세그먼트 단위로 하나씩 읽고 재생하는 과정을 반복한다[5]. 서비스주기는 하나의 세그먼트가 재생되는 시간이다. 다중 스트림들을 서비스하기 위하여 각 스트림 당 하나의 세그먼트씩을 서비스주기내에 모두 읽어들여야 한다. 서비스주기는 세그먼트가 소모(재생)되는 시간보다 작아야 한다. 서비스주기를 길게 하려면 세그먼트 크기를 크게 해야 한다. 서비스주기가 세그먼트 소모시간 보다 길어지면 스트림 재생시 깜빡거림(hiccup) 현상이 일어난다. 따라서, 서비스주기는 세그먼트 크기에 의하여 결정되고 식 (1)을 만족해야 한다. 이후부터 주로 사용될 기호들중에서 T_{period}는 서비스주기, S는 세그먼트 크기, R_{med}는 스트림 소모율, R_{disk}는 디스크 전송률, $D_{latency}$는 디스크내에서의 탐색지연시간, m은 서비스주기동안 하나의 디스크에서 검색될 수 있는 세그먼트 개수를 의미한다.

$$m * \left(\frac{S}{R_{disk}} + D_{latency}\right) \le T_{period} \le \frac{S}{R_{med}} \quad (1)$$

각 서비스주기동안 각 스트림 당 하나의 세그먼트들이 버퍼로 읽혀지고 읽혀진 세그먼트는 재생된다.

단일버퍼링(single buffering)을 가정하면 세그먼트가 버퍼로 읽혀지는 동안 동일버퍼에서 데이터 소모도 동시에 일어난다[5]. 따라서, 버퍼에 데이터가 읽혀지는 속도는 $R_{disk} - R_{med}$가 된다. T(i)가 서비스주기 동안 어떤 스트림 i에 할당된 시간슬롯이라고 한다면 그 스트림을 위하여 요구되는 버퍼량은 $(R_{disk} - R_{med})*T(i)$이고 이는 세그먼트의 크기 S와 같다.

현재 m개의 스트림들이 서비스중일 때 필요로 되는 버퍼용량은 식 (2)처럼 표현될 수 있다[5].

$$B = \sum_{i=1}^{m} [(R_{disk} - R_{med}) * T(i)] = \sum_{i=1}^{m} [R_{med} * (T_{period} - T(i))] = m * S \quad (2)$$

식 (2)에서 볼 수 있는 것처럼 단일버퍼링은 버퍼용량을 적게 요구하는 대신 버퍼로 읽어들이는 데이터량이 스트림 소모율 만큼 감소되고 있슴을 알 수 있다. 또한, 단일버퍼링은 각 서비스주기에서 스트림의 검색순서를 유지하여야 스타베이션(starvation) 현상이 발생하지 않는다[5].

더블버퍼링(double buffering)은 세그먼트가 버퍼로 읽혀지는 동안 다른 버퍼에서 데이터 소모가 일어난다. 따라서, 버퍼에 데이터가 읽혀지는 속도는 디스크 전송속도 R_{disk}를 유지할 수 있다. $T(i)$가 서비스 주기동안 어떤 스트림i에 할당된 시간슬롯이라고 한다면 그 스트림을 위하여 요구되는 버퍼량은 $2*R_{disk}*T(i)$이고 이것은 세그먼트 크기 S의 2배이다. 현재 m개의 스트림들이 서비스중일 때 필요로 되는 버퍼용량은 식 (3)처럼 표현될 수 있다.

$$B=\sum_{i=1}^{m}\left[2*R_{disk}*T'(i)\right]=\sum_{i=1}^{m}\left[2*R_{med}*T_{period}\right]=2*m*S \qquad (3)$$

식 (3)에서 볼 수 있는 것처럼 더블버퍼링은 버퍼용량을 많이 요구하는 대신 세그먼트를 버퍼로 읽어들이는 속도는 디스크 전송속도를 그대로 유지할 수 있음을 알 수 있다. 또한, 더블버퍼링은 각 서비스주기에서 스트림의 검색순서를 유지하지 않아도 된다.

다중디스크 환경에서 비디오스트림의 저장 및 검색정책과 관련하여 기존에 연구된 주요 결과들은 RR(Round Robin), VSP(Virtual Sequential Pumping) 등이 있다[1,6]. 각 정책들은 비디오스트림 Vi를 세그먼트들 V_i^0, V_i^1, V_i^2, ... 으로 분할하여 각 세그먼트들을 X개의 디스크들 D_0, D_1, D_2,..., D_{X-1}상에 분산 배치한다.

RR 정책에서 임의의 비디오 세그먼트 V_i^j가 X개의 디스크들 D_0, D_1, D_2, ..., D_{X-1}상에 배치될 때 디스크 D(i+j)modX 의 임의의 위치에 배치된다[6]. RR정책은 각 디스크에서 디스크헤드의 임의이동에 기반하여 다중 세그먼트 서비스를 하므로 세그먼트 검색 사이의 탐색시간 등으로 디스크 대역폭을 낭비한다. 그러나, 스트림 요구에 대한 응답은 즉각적으로 이루어진다.

VSP 정책은 비디오스트림이 연속적인 세그먼트들로 이루어지고 비디오스트림의 재생은 연속적인 세그먼트들을 검색한다는 특징을 이용하여 디스크 대역폭을 절약한다. 각 디스크 영역을 동일한 수의 블록데이터를 갖고 동일한 디스크 전송률을 갖는 서브영역 즉, 존(zone) 들로 분할하고 비디오스트림의 세그먼트들을 특정 디스크의 특정 존에 배치한다[1]. 디스크 배열에서 X개의 디스크 D_0, D_1, D_2, ..., D_{X-1}이 있고 각 디스크는 Y(=X+1)개의 존들 Z_0, Z_1, Z_2, ..., Z_{Y-1}로 나누어져 있다고 할 때 임의의 비디오 세그먼트 V_i^j의 전역 세그먼트 번호가 g라면 이 세그먼트는 디스크 $D_{g \bmod X}$의 존 $Z_{g \bmod Y}$에 배치된다. 전역세그먼트 번호는 세그먼트들에 부여된 일련번호인데 다른 스트림들의 세그먼트들도 고려하여 순서대로 부여된 세그먼트 번호이다. 예를 들어 스트림 V_0와 V_1의 세그먼트들에 대한 전역 세그먼트 번호를 구할 때 스트림 V_0가 6개의 세그먼트들 V_0^0, V_0^1,V_0^2,V_0^3, V_0^4,V_0^5 로 이루어질 경우 스트림 V_1의 첫 번째 세그먼트 V_1^0의 전역세그먼트번호는 7이 된다. 데이터 검색 시 다중디스크 환경의 모든 디스크들의 디스크헤드들은 서비스주기마다 존 단위로 동기화(synchronization)를 유지하면서 안쪽 존부터 바깥쪽 존으로 순차적 이동을 하면서 요구 세그먼트들을 읽는다. 특정 서비스주기동안에 각 디스크들은 특정 존 내의 세그먼트들만을 검색하므로 탐색지연시간은 RR 정책의 경우보다 짧아진다. RR 정책의 탐색지연을 $D_{latency}$라고 했을 때 VSP의 탐색지연 $Z_{latency}$는 식 (4)와 같이 표현될 수 있다.

$$Z_{latency} \approx \frac{D_{latency}}{Y} \qquad (4)$$

(Y : 단일 디스크에서의 존 갯수)

$Z_{latency}$는 $D_{latency}$보다 짧으므로 VSP 정책의 디스크 대역폭은 RR 정책에 비해서 절약되고 절약된 디스크 대역폭만큼 더 많은 스트림들을 서비스할 수 있다. 그러나, VSP 정책에서 시작지연은 RR 정책보다 더 길다. RR 정책에서는 디스크헤드의 임의이동을 기반으로 함으로 요구 스트림의 시작 세그먼트를 즉각적으로 읽을 수 있지만 VSP 정책에서는 디스크헤드가 다른 디스크들의 헤드들과 존 단위의 동기화를 유지하면서 요구 스트림의 시작 세그먼트가 있는 존으로 올 때 까지 기다려야 한다. 또한, 요구 스트림들의 세그먼트들이 특정 존 영역으로 집중될 경우 시작지연은 더 길어질 수 있다.

3. 새로운 정책

본 절에서는 RR 정책과 VSP 정책을 보완할 수 있는 새로운 정책 POR(Priority-Oriented-Retrieval) 검색정책을 제안한다.

3.1 POR 검색정책 및 알고리즘

POR 검색정책은 VSP 배치정책에 의하여 배치

된 스트림들을 검색할 때 스트림들의 세그먼트들에 우선순위(priority)를 부여하여 우선순위 순으로 세그먼트들을 읽음으로써 스트림들의 시작 세그먼트들을 즉각적으로 읽을 수 있게 하고 세그먼트 검색 사이의 탐색지연시간을 줄임으로써 디스크 대역폭을 절약할 수 있다. 세그먼트의 우선순위는 순서쌍(Time_Priority, Position_Priority)가 낮은 순으로 높은 우선순위가 부여된다. 즉, Time_Priority가 낮을수록 높은 우선순위를 갖고 동일한 Time_Priority에서는 Position_Priority가 낮을수록 높은 우선순위를 갖는다. 여기서, Time_ Priority 는 스트림의 세그먼트들에 부여되는 번호인데 스트림내에서 그 세그먼트의 검색순서를 의미한다. Position_Priority는 세그먼트가 저장된 존 번호이다. 그림1은 POR알고리즘을 나타내고 있다.

그림 2의(a)는 3개의 디스크 D_0, D_1 D_2가 있고 각 디스크는 3개의 존 Z_0, Z_1, Z_2로 나누어졌을 때3개의 스트림들 V_0, V_1, V_2 들이 VSP배치정책을 이용하여 디스크의 존 상에 배치된 상태를 나타내고 있다. 그림2의 (b)는 각 디스크의 각 존에 대응되는 큐의 상태를 나타내고 있다. 큐 Q(i,j)는 디스크 D_i의 존 Z_j에 대응되는 큐임을 나타낸다. 각각의 큐 안에는 해당 디스크의 해당 존에서 읽혀져야 할 세그먼트 정보들과 각 세그먼트들의 우선순위 정보가 들어 있다. Q(0,0)의 경우 세그먼트 V_0^0는 우선순위 순서가 (0,0)임을 의미한다. 즉, Time_Priority가 0이고

```
Algorithm POR
Input:
   Q(i,j) // Queues for segment requests being serviced in
   zone Zj of disk Di, j = 0, 1, .2, ... , Y-1//
 Variable:
   Tcurr  // the current service period number //
   Y  // the number of zones in disk //
Begin
    While (Current Service Period)
        for (j ← 0 ; j≤Y-1 ; j++)
                service the Segment Requests of which
                Time_Priority equal to Tcurr in Q(i,j) ;
  end for
  Current Service Period←Next Service Period;
End While
End
```

그림 1. POR 알고리즘

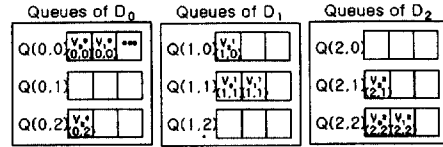
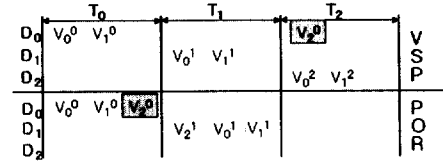
(a) 스트림 배치상태

(b) 디스크의 큐 상태

(c) 스트림 검색서비스

그림 2. POR예제

Position_Priority는 0임을 나타낸다. 그림2의 (c)는 3개의 스트림들 V_0, V_1, V_2 들에 대한 요구가 발생했을 때 POR 검색정책과 VSP 검색정책이 각 스트림의 세그먼트들을 서비스하는 상황을 나타내고 있다. VSP 검색정책은 존을 기준으로 디스크헤드들 사이의 동기화를 유지하여야 하므로 서비스주기 T_0동안에 존 Z_0에서 스트림 V_0와 V_1의 세그먼트들 V_0^0, V_1^0을 검색하고 디스크 대역폭의 여분이 있다고 할 지라도 스트림 V_2의 시작 세그먼트 V_2^0를 서비스할 수 없다. 반면, POR 검색정책에서는 세그먼트의 우선순위 중 Time_Priority가 현재의 서비스주기 번호와 동일한 세그먼트를 우선 검색하고, 동일한 Time_Priority를 갖는 세그먼트가 여러 개 있을 경우 Position_Priority가 낮은 순서대로 세그먼트들을 검색하므로 서비스주기 T_0동안에 스트림 V_0와 V_1의 세그먼트들 V_0^0, V_1^0를 검색하고 디스크 대역폭의 여분이 있으면 스트림 V_2의 시작 세그먼트 V_2^0를 서비스할 수 있다. 즉, VSP 검색정책은 스트림 V_2의 서비스를 서비스주기 T_2에서 시작하지만 POR은 서비스주기 T_0에서 시작할 수 있다. 여기서, POR 정책은 각 서비스주기동안 스트림들의 서비스 순서가 달라지지만 더블버퍼링을 적용할 것이므로 문제가 되지 않는다. POR 검색정책은 동일한 Time_Priority를 갖는 세그먼트들에 대해서 Position_Priority가 낮은 순서대로 세그먼트들을 검색하므로 세그먼트들 사이의 탐색지연시간이 $D_{latency}$ 대신에 $Z_{latency}+\beta$가 되고

시작지연은 임의접근시의 시작지연과 비슷하게 단축된다. POR 검색정책은 전통적인 디스크 스케줄링 정책중의 하나인 SCAN 개념과 비슷하지만 비디오 스트림의 시간제약 특성을 만족시킨다.

3.2 RR, VSP, POR의 탐색지연 분석

앞에서 살펴 본 바와 같이 RR 기법의 탐색지연은 $D_{latency}$이다. VSP 기법의 경우 존 내에서 세그먼트 검색사이의 탐색지연은 $Z_{latency}$이지만 보다 자세하게 분석해 보면 $Z_{latency}+\alpha$로 간주해야 한다. VSP 기법은 존 Z_0부터 존 Z_{Y-1}까지 서비스가 이루어진 다음 다시 존 Z_0부터 서비스를 해야 하므로 각 서비스주기 내에 디스크헤드의 되감기 시간을 고려해 주어야 한다. 식 (5)에서 보여주는 바와 같이 $Z_{latency}+\alpha$는 되감기 시간이 고려되었을 때의 탐색지연시간이다. 식 (5)에서 T_{rewind}는 디스크헤드의 되감기 시간이고 m은 서비스주기동안 검색될 수 있는 세그먼트 개수이다.

$$Z_{latency}+\alpha=\frac{D_{latency}}{Y}+\frac{T_{rewind}}{m} \quad (5)$$

POR 검색정책에서의 탐색지연은 $Z_{latency}+\beta$가 되는데 β값의 의미를 예제를 통하여 알아보기로 한다. 그림3은 각 디스크상에 우선순위가 (a, b_1), (a, b_2), (a, b_3), (a, b_4) (단, $b_1 \le b_2 \le b_3 \le b_4$)인 4개의 세그먼트들 S_1, S_2, S_3, S_4가 배치되어 있고 각 디스크는 4개의 존들로 나누어지고 각 존 들은 3개의 실린더로 이루어진 상태를 나타내고 있다.

그림3에서 점선 화살표는 세그먼트 검색순서를 의미한다. 그림3의 (a), (b), (c)는 POR 검색정책의 세그먼트 검색 양상을 보여주고 있고 (d)는 VSP 검색정책의 세그먼트 검색 양상을 보여주고 있다. 디스크 (a), (b), (c) 에 있는 4개의 세그먼트들을 검색하는 시간은 동일하고 식 (6) 처럼 표현할 수 있다. 식 (6)은 디스크 (c)를 기준으로 유도된다. 여기서, T_{move}는 존 사이의 이동 시간이고 S_{trans}는 세그먼트 전송시간이다.

그림 3. 우선순위 지향 검색의 순차처리 성질

$$\begin{aligned}&(Z_{latency}+T_{trans})+(Z_{latency}+T_{trans}+T_{move})+(Z_{latency}+T_{trans}+Z_{move})+(Z_{latency}+T_{trans}+T_{movet})\\&=4\bullet(Z_{latency}+T_{trans})+3\cdot T_{move}\end{aligned} \quad (6)$$

POR 검색에서 서비스주기 T_{period}동안 m개의 세그먼트들을 서비스한다면 식 (6)을 일반화하여 식 (7)과 같이 나타낼 수 있다. 다음 서비스주기를 시작하기 전에 디스크헤드는 처음 존에 위치되어야 하기 때문에 서비스주기내에 디스크헤드의 되감기 시간이 필요하다. 식 (7)에서 T_{rewind}는 되감기 시간을 나타낸다.

$$T_{period}\ge m*(Z_{latency}+T_{trans})+(Y-1)*T_{move}+T_{rewind} \quad (7)$$

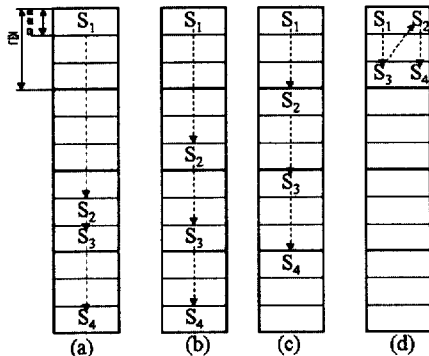
식 (7)에 의해서 β값은 식 (8)과 같이 나타낼 수 있다.

$$\beta=\frac{(Y-1)*T_{move}+T_{rewind}}{m} \quad (8)$$

따라서, 식 (9)와 같은 결과를 유도할 수 있고 식 (9)는서는 RR, VSP, POR정책의 탐색지연을 비교하고 있다.

$$Z_{latency}+\alpha < Z_{layency}+\beta < D_{latency} \quad (9)$$

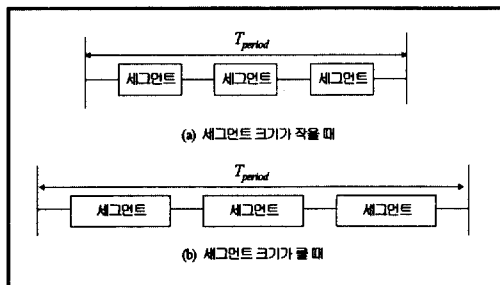
3.3 적응형 버퍼관리 방법

이 정책은 단일버퍼링에서 요구되는 버퍼량으로 더블버퍼링을 적용했을 때에도 버퍼제약으로 인한 서비스 가능한 스트림 수의 한계를 최소화시키기 위한 목적을 갖는다. 일반적으로 세그먼트 크기가 작으면 적은 버퍼량을 요구하지만 서비스 주기가 짧아져서 서비스 주기동안 읽을 수 있는 세그먼트 수는 적어질 것이라고 기대할 수 있다. 마찬가지로, 세그먼트 크기가 크면 많은 버퍼량을 요구하지만 서비스 주기가 길어져서 서비스 주기동안 많은 세그먼트들을 검색할 수 있을 것이라고 기대할 수 있다. 그러나, 세그먼트가 커짐에 따라 세그먼트 전송 시간도 증가

하게 되고 세그먼트 검색 사이의 탐색지연시간이 세그먼트 전송 시간에 비하여 무시될 수 있는 상황이 발생한다. 이 때, 세그먼트 크기를 계속 증가시켜서 서비스 주기가 계속 길어진다 할 지라도 그 서비스 주기동안 읽을 수 있는 세그먼트들의 수는 디스크전송률/스트림소모율 이상 증가하지 않는다. POR 검색정책에서 디스크 처리율을 최대로 하는 세그먼트 크기 S_{max}는 식 (1)로부터 식 (10)처럼 계산할 수 있다.

$$S_{max} = \frac{Z_{latency} + \beta}{\left(1/\left(\lfloor R_{disk}/R_{med} \rfloor * R_{med}\right) - 1/R_{disk}\right)} \tag{10}$$

세그먼트 크기가 S_{max}이상이 되면 세그먼트 크기가 커져도 서비스 주기동안 읽을 수 있는 세그먼트들의 수는 증가하지 않는다. 비슷한 개념으로, 세그먼트 크기가 작아짐에 따라 서비스 주기는 짧아지는 반면 그와 비례해서 읽어야 할 세그먼트들의 크기도 작아지므로 서비스 주기동안 읽을 수 있는 세그먼트들의 수는 세그먼트 크기가 작아지는 것과 비례해서 감소하지는 않는다. 그림4는 이러한 개념을 도식적으로 나타내고 있다.

크기가 S_{max}인 세그먼트가 재생되는 시간동안 단일 디스크에서 M_{max}개의 세그먼트들을 읽을 수 있다고 하면 X개의 디스크들로 구성된 다중 디스크환경에서의 최대 동시허용 스트림 수는 M_{max}*X(X는 디스크수)가 되지만 이것은 디스크 대역폭만을 고려하여 계산된 것이고 요구 버퍼량은 2*S_{max}*M_{max}*X (더블버퍼링 일 때)가 된다. 제공된 버퍼용량이 이 보다 적으면 디스크 대역폭이 지원할 수 있는 동시 허용 스트림 수보다 더 적은 스트림들을 지원한다.

따라서, 제공되는 디스크 대역폭과 버퍼용량을 동시에 고려하여 세그먼트 크기를 결정하는 것이 바람직하다. 주어진 버퍼크기 B_{give}과 디스크 대역폭 R_{disk}를 동시에 고려하여 휴리스틱하게 조절되는 세그먼트 크기 S_{adjust}는 식 (11) 처럼 계산할 수 있다.

$$S_{adjust} = S_{max} - \frac{2 \cdot S_{max} \cdot M_{max} \cdot X - B_{give}}{M_{max} \cdot X} \tag{11}$$

4. 실험 및 결과 분석

실험을 위한 몇 가지 가정을 설정하였다. 첫째, 다중디스크 환경의 각 디스크헤드들은 하드웨어적으로는 비동기적으로 움직일 수 있다. 현재 상용화된 RAID 시스템의 각 디스크들은 독립적인 디스크 컨트롤러들이 장착되어 있는 것이 일반적이므로 이러한 가정은 무리가 없을 것으로 판단된다[7]. 둘째, 임의의 수의 비디오스트림들은 관련 배치정책을 기반으로 다중 디스크상에 배치되어 있다고 가정하고 랜덤 함수를 사용하여 요구 스트림의 시작 세그먼트가 있는 디스크와 존이 결정된다. 셋째, 요구 스트림의 도착 시간은 평균이 3초인 Poisson분포를 따른다. 넷째, 시작지연시간 측정시 동시에(concurrently) 허용 가능한 스트림 수까지의 평균 시작지연시간만을 고려한다. 실험을 위한 디스크 모델은 Seagate Wren8 ST41650N[5]를 사용한다. 비디오스트림들은 MPEG-I으로 압축되어 비교 모델들의 관련 배치정책을 적용하여 다중 디스크상에 분산 배치되어 있다고 가정한다. 시뮬레이션은 다중디스크 환경 및 버퍼용량을 변화시켜 가면서 동시 허용 스트림 수 및 평균 시작지연시간을 측정한다.

그림5는 제공되는 버퍼 크기를 무한대로 하고 다중디스크 환경의 디스크 수를 16으로 하였을 때 세그

그림 4. 서비스 주기와 세그먼트 크기와의 관계

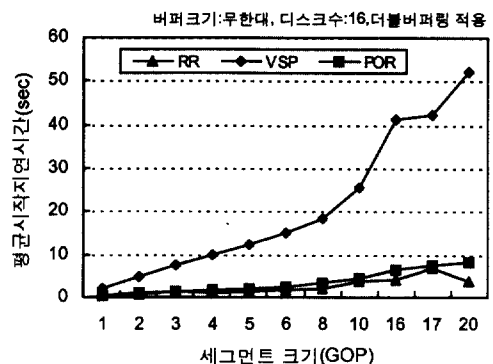



그림 5. 최대 허용 스트림 수까지의 평균 시작지연

먼트 크기당 평균 시작지연시간을 보여주고 있다. 세그먼트 크기는 MPEG의 GOP(Group Of Pictures)크기의 배수이다. POR은 우선순위에 기반한 세그먼트 검색으로 인하여 직접접근 효과가 발생함으로써 시작지연시간이 VSP보다 짧고 RR과 거의 비슷함을 알 수 있다. 전체적으로 세그먼트 크기가 작을 때는 세그먼트 크기가 클 때에 비하여 평균 시작지연시간이 짧다. 세그먼트 크기가 작으면 세그먼트를 읽는 시간도 단축되고 요구 스트림의 시작 세그먼트에 도착하는 시간도 짧아지게 된다.

그림 6은 제공되는 버퍼 크기를 무한대로 하고 다중디스크 환경의 디스크 수를 16으로 하였을 때 세그먼트 크기당 최대의 동시 허용 스트림 수를 보여주고 있다. 디스크 처리율을 최대로 하는 세그먼트 크기(이때, 최대로 서비스 가능한 스트림 수는 224개이다)는 RR 정책의 경우 20이고 VSP 정책은 15, POR 정책은 16임을 알 수 있다. 이때, 세그먼트 크기가 작을수록 버퍼용량을 절약할 수 있으므로 최대 스트림 수를 지원하기 위하여 요구되는 세그먼트 크기가 작을수록 좋다. 따라서 RR보다는 VSP나 POR이 좋은 정책임을 알 수 있다.

그림 7은 세그먼트 크기가 8일 때 디스크 수당 허용 스트림 수를 보여주고 있다. VSP와 POR정책이 타 정책에 비하여 7% 정도 더 많은 스트림들을 허용하고 있슴을 알 수 있다.

그림 8은 적응형 버퍼관리 기법의 효과 즉, 주어진 버퍼용량을 고려하여 세그먼트 크기를 결정하였을 경우의 효과를 나타내고 있다. 여기서는 디스크 처리율을 최대로 할 수 있는 세그먼트 크기를 가정하고 있다. 적응형 버퍼 관리기법은 단일버퍼링에서 요구

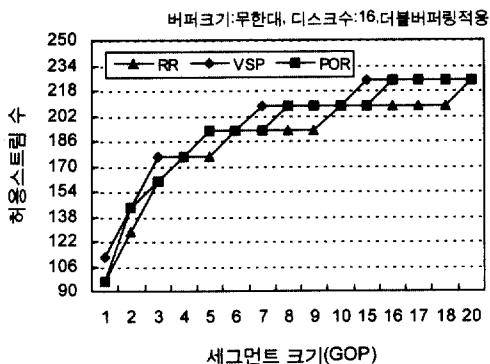



그림 6. 세그먼트 크기 당 허용 스트림 수

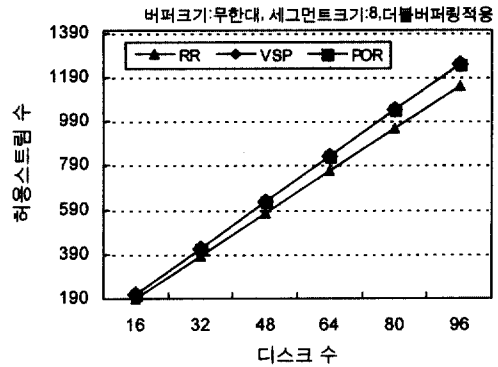



그림 7. 디스크 수 당 허용 스트림 수(세그먼트 크기 8)

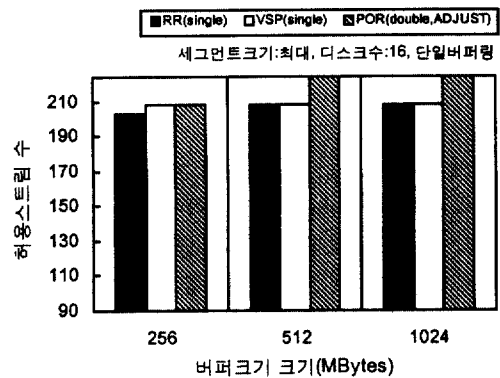



그림 8. 적응형 버퍼관리의 효과

되는 버퍼량으로 더블버퍼링을 적용할 때에도 서비스 가능한 스트림들의 수를 많게 하기 위한 정책이다. 그림 8에서 보여주듯이 버퍼용량이 256MB이면 RR 정책에서 단일버퍼링을 사용한다 할 지라도 버퍼제약으로 인하여 디스크 대역폭이 지원할 수 있는 최대 스트림 수를 제공할 수 없다. 반면, 주어진 버퍼용량과 디스크 대역폭을 고려하여 세그먼트 크기를 결정하면 버퍼제약으로 인한 서비스 가능한 스트림 수에 대한 한계는 해소되고 디스크 대역폭도 최적으로 활용할 수 있는 상황이 되므로 더블버퍼링을 사용하면서도 RR 정책보다 더 많은 스트림들을 서비스할 수 있음을 알 수 있다.

5. 결 론

본 논문에서는 비디오 저장시스템에서 주어진 디스크 대역폭과 버퍼를 효율적으로 활용하여 비디오 서버로의 서비스요구에 대한 시작지연시간을 짧게

유지할 뿐만 아니라 동시에 서비스 가능한 스트림들의 수를 많게 하기 위한 정책들을 제안하였다.

서비스요구의 시작지연을 단축시키기 위한 방안으로 우선순위 지향 검색정책을 제안하였다. POR 검색정책은 스트림들의 세그먼트들에 우선순위를 부여하여 우선순위 순으로 세그먼트 검색을 함으로써 디스크헤드들 사이의 존 단위의 동기화가 해제되고 임의이동 효과가 발생함으로써 기존 VSP 정책에 비하여 시작지연시간이 단축될 수 있다. POR 검색정책은 비디오스트림의 시간제약 요구사항을 고려하기 위하여 어떤 스트림의 세그먼트들에 대한 우선순위는 타 스트림들과 독립적으로 결정됨으로써 실행시간 시 우선순위 부여에 의한 시간적 부담이 제거된다.

보다 많은 스트림들을 서비스하기 위한 정책으로 두가지 방안들을 제안하였다. 첫째는 POR 검색정책에 의하여 디스크헤드의 순차적 이동 효과를 유도함으로써 세그먼트 검색 사이의 탐색지연시간을 단축시키고 디스크 대역폭을 절약하였다. 절약된 디스크 대역폭 만큼 추가적인 스트림들을 서비스할 수 있다. 둘째는 버퍼의 효율적 활용 방안이다. POR 검색정책이 더블버퍼링을 기반으로 수행되므로 타 정책에 비하여 더 많은 버퍼를 필요로 한다. 이를 보완하기 위하여 제공된 버퍼용량과 디스크 대역폭을 고려하여 최적의 세그먼트 크기를 결정할 수 있는 휴리스틱 알고리즘을 제안하였다. 이 알고리즘은 단일 버퍼 관리 기법에서 요구되는 버퍼량으로도 POR 정책을 효율적으로 지원할 수 있음을 알 수 있었다.

참 고 문 헌

[1] Gin-Kou Ma, Chiung-Shien Wu, Mei-Chian Liu, and Bao-Shuh P. Lin, Efficient Real-time Data Retrieval Through Scalable Multimedia Storage, *Proceedings of ACM Multimedia 97*, 1997, pp165-172.

[2] Oge Marques and Borko Furht, Issues in Designing Contemporary Video Database Systems, *Proceedings of IASTED International Conference IMSA'99*, Oct, 1999, pp43-47.

[3] D.James Gemmel, Harrick M.Vin, Dilip D. Kandlur, P.VenkatRangan, Multimedia Storage Server: A Tutorial and Survey, *IEEE Computer*, Vol. 28, No.5, May, 1995, pp40-49.

[4] Steven Berson, Shahram Ghandeharizadeh, Staggered Striping in Multimedia Information Systems, *Proceedings of ACM SIGMOD*, 1994, pp79-90.

[5] Huang-Jen Chen, Thomas D.C. Little, Storage Allocation Policies for Time-Dependent Multimedia Data, *IEEE Transactions on Knowledge and Data Engineering*, Vol.8, No.5, Oct, 1996, pp855-864.

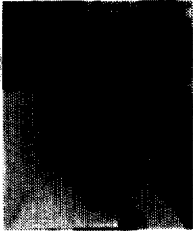
[6] Taeck-Geun Kwon, Sukho Lee, Data Placement for Continuous Media in Multimedia DBMS, *Proceedings of IEEE Multimedia DBMS*, August 28-30, 1995, pp110-117.

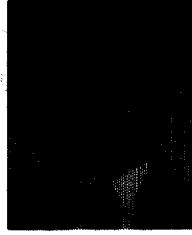
[7] Patterson, D., Gibson, G., and Katz, R., A case for Redundant Arrays of Inexpensive Disks (RAID), *Proceedings of SIGMOS88*.

[8] A. Alan and B. Prisker, Simulation with Visual SLAM and AweSim, *Systems Publishing Corporation*, 1997.

[9] B.Ozden, R.Rastigi and A.Siberschatz, Disk Striping in Video Server Environments, *Data Engineering*, 18(4), 1995, pp4-16.

[10] S.Park, KeunHyung.Kim, Efficient Storage and Retrieval Scheme of Video Streams in VOD Server, *Proceedings of IASTED International Conference IMSA'99*, 1999, pp163-169.

[11] M.S. Chen, D. Kandlur, P. Yu, Support for Fully Interactive Playout In Disk-Array-Based Video Server, *Proceedings of ACM Multimedia 94*, 1994, pp391-398.

김 근 형

1990. 2 서강대학교 컴퓨터학과 졸업(학사)
1992. 2 서강대학교 컴퓨터학과 졸업(석사)
2001. 2 서강대학교 컴퓨터학과 졸업(박사)
1992. 1~1994. 2 현대전자 소프트웨어연구소
2001. 9~현재 제주대학교 경영정보학과 전임강사
E-mail : khkim@cheju.cheju.ac.kr

박 석

1978. 2 서울대학교 계산통계학과 졸업(학사)
1980. 2 한국과학기술원 전산학과 졸업(석사)
1983. 8 한국과학기술원 전산학과 졸업(박사)
1983. 9~현재 서강대학교 컴퓨터학과 교수
1989. 12~1991. 1 University of Virginia 방문교수
1999. 5~2000. 12 정보통신부 정보통신 우수대학원 심의위원